

Multimed Tools Appl (2017) 76:9017–9032  
DOI 10.1007/s11042-016-3508-x



# Increasing anti-spoofing protection in speaker verification using linear prediction

Artur Janicki<sup>1</sup>

Received: 15 September 2015 / Revised: 27 January 2016 / Accepted: 1 April 2016 /

Published online: 16 April 2016

© The Author(s) 2016. This article is published with open access at Springerlink.com

**Abstract** This article addresses the problem of anti-spoofing protection in an automatic speaker verification (ASV) system. An improved version of a previously proposed spoofing countermeasure is presented. The presented method is based on the analysis of linear prediction error that results from both short- and long-term prediction of the input speech signal. It was observed that non-natural speech signals, i.e., synthetic or converted speech, were predicted in a different way than genuine speech. Therefore, in contrast to the classical linear prediction analysis, where usually only the prediction coefficients are analyzed, in the proposed approach the residual (error) signals were examined. During this analysis, 23 various prediction parameters were extracted, such as the energy of the prediction error, prediction gains and temporal parameters related to the prediction error signals. Various binary classifiers were researched to separate human and spoof classes, however the support vector machines with radial basis function (SVM-RBF) yielded the best results. When tested on the corpora provided for the ASVspoof 2015 Challenge, the proposed countermeasure returned better results than the previous version of the algorithm and, in most of the cases, the baseline spoofing detector based on the local binary patterns (LBP). It is hoped that the proposed method can be part of a generalized spoofing countermeasure helping to increase security of ASV systems.

---

The calculations described in the article were made in the Interdisciplinary Centre for Mathematical and Computational Modelling (ICM) of the University of Warsaw (computational grant No. G46-2).

---

✉ Artur Janicki  
A.Janicki@tele.pw.edu.pl

<sup>1</sup> Division of Cybersecurity, Institute of Telecommunications, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland

**Keywords** Speaker verification · Spoofing · Linear prediction · Local binary patterns · SVM-RBF

## 1 Introduction

User authorization based on a human voice, often referred to as speaker verification, is becoming more and more popular. Automatic speaker verification systems (ASVs) are used to authorize access, e.g., to a mobile phone, or to authenticate customers over a phone line. The latter application is already used, e.g., by banks such as Bank Smart in Poland [5], or public institutions, such as the Australian Taxation Office [4].

Similar to other biometric modalities, systems based on voice are also prone to spoofing, i.e., attacks to get illegitimate access. In the context of ASV systems, spoofing is realized by presenting artificial or manipulated speech, which can be generated using, e.g., speech synthesis or voice conversion algorithms. Since these techniques are becoming more easily available and are constantly improving their quality, they have begun to pose a major threat to ASV systems. Quite recently, researchers have started to investigate how much ASV systems are prone to spoofing using various methods. Various researchers have worked on assessing the threat caused by imitators [13, 23], speech synthesis [26, 35], converted speech [22, 39] or replay of previously acquired recordings [3, 24]. In parallel, much effort has been invested in work on various spoofing countermeasures. These algorithms can be either dedicated to a given attack or can be generally applicable. A thorough review of spoofing methods and their countermeasures can be found in [40].

In [19] a novel spoofing countermeasure was proposed. It was based on analysis of linear prediction error and used a logistic classifier as a detection algorithm. The idea of this algorithm was inspired by the fact that synthetic or converted voice is quite likely to be either very easily predicted, if generated with a simplified acoustic model, or very difficult to predict, if any artifacts in the signal are present. Experiments conducted using the datasets provided by the organizers of the first ASV Spoofing and Countermeasures Challenge ASVspooF 2015 [41] showed that the proposed method was able to detect spoofing effectuated with voice conversion and speech synthesis. When testing spoofing detection independently from an ASV system, it yielded an equal error rate (EER) of less than 9 % for the Development corpus, while the baseline method based on local binary patterns (LBP) resulted in an EER higher than 14 %.

The work described in this article is the continuation of the author's research aiming to contribute to the speech community's efforts to find efficient anti-spoofing methods for ASV systems. In the current study, a new version of the spoofing detection algorithm was proposed. It uses an extended set of parameters and employs a new, more powerful classifier. The performance of this anti-spoofing protection is compared with the previous version of this algorithm [19] as well as with the results of the detector based on LBPs, which has been efficient in other studies [2].

In this article, first the state-of-the-art in spoofing countermeasures is summarized and the basics of linear prediction theory are recalled. Then, in Section 3 the proposed countermeasure is presented and compared with its previous version. In Section 4 the experimental set-up is described. In Section 5 the results are presented and compared with the results from

the previous version of the algorithm and the LPB-based countermeasure. Finally, Section 6 concludes the paper and presents perspectives for future work.

## 2 Previous work

### 2.1 Spoofing countermeasures for ASV systems

One group of countermeasures exploits prior knowledge about the origin of the spoofing attack. For example, some algorithms detect artifacts typical of speech synthesis, such as simplification of F0 contours [27]. In [1] the authors proposed an algorithm that was based on measuring the pair-wise distance (PWD) between spectral parameters (such as linear prediction coefficients or Mel-cepstral coefficients) in consecutive frames. The authors claimed that voice conversion decreases PWD values and, as a consequence, changes PWD distributions. They compared speaker-dependent PWD distributions between genuine and converted speech using speech data from the NIST'06 database and the NIST SRE protocol. The authors showed that the proposed countermeasure was able to lower the EER from more than 30 % to below 3 %.

Countermeasures dedicated to detecting replay attacks often try to identify unexpected channel artifacts indicative of recording and replaying. Such algorithms were reported in [37], for which the EER of a baseline GMM-UBM system was shown to decrease from 40 % to 10 % with active countermeasures. Another replay countermeasure aimed at detecting far-field recordings, which are unlikely in natural access scenarios where the speaker is usually close to the microphone [36].

Not many algorithms have so far been reported that claim to be more generally applicable and less dependent on prior knowledge of the attack. For example, one group of such methods exploits the fact that many speech synthesis and voice conversion algorithms disturb the natural phase of the speech signal. In [9] the authors challenged GMM-UBM and SVM-GMM speaker verification systems with genuine and synthesized speech originating from the WSJ corpus. They showed that by using relative phase shift (RPS) features it was possible to decrease the EER from over 81 % to less than 3 %. Unfortunately, the method proposed was vocoder-dependent. Similarly, phase information was successfully used in detecting converted speech in [38].

Another generalized method was presented in [2]. It was based on the LBP analysis of speech cepstograms and was inspired by an original application to image texture analysis [28]. In this approach LBP analysis was applied to a Mel-scaled cepstrogram with appended dynamic features. The authors claimed that modifications made through spoofing disturb the natural 'texture' of the speech signal. Experimental results showed that the LBP-based textrogram analysis was very effective in detecting spoofing trials generated using speech synthesis (EERs of below 1 %), but it was less effective in detecting those originating from voice conversion (EER in the order of 7 %).

### 2.2 Linear prediction theory

Linear prediction theory dates back to the 1940s [33]. It has been used not only in speech processing, but also in neuroscience and geology [25]. As for speech processing, linear

prediction was originally used in speech coding, where a technique called linear prediction coding (LPC) was developed. It consists of calculating the so-called prediction coefficients  $a_i$  so that in a frame (e.g., 20ms long) of the input speech signal each signal sample  $x(n)$  can be efficiently predicted using the  $p$  preceding samples by the value  $\hat{x}(n)$  calculated as follows:

$$\hat{x}(n) = \sum_{i=1}^p a_i x(n-i) \quad (1)$$

The difference between the original signal  $x(n)$  and the predicted signal  $\hat{x}(n)$  is called the residual signal or the prediction error  $e(n)$ . The value  $G_p$  defined as:

$$G_p = \frac{E_x}{E_e} \quad (2)$$

where  $E_x$  is the energy of signal  $x(n)$  and  $E_e$  is the energy of the prediction error  $e(n)$ , is called the prediction gain. The higher the gain, the better the signal will be predicted, so this means that prediction coefficients  $a_1..a_p$  are able to efficiently model the speech signal within a frame. As a consequence, this will enable a better compression.

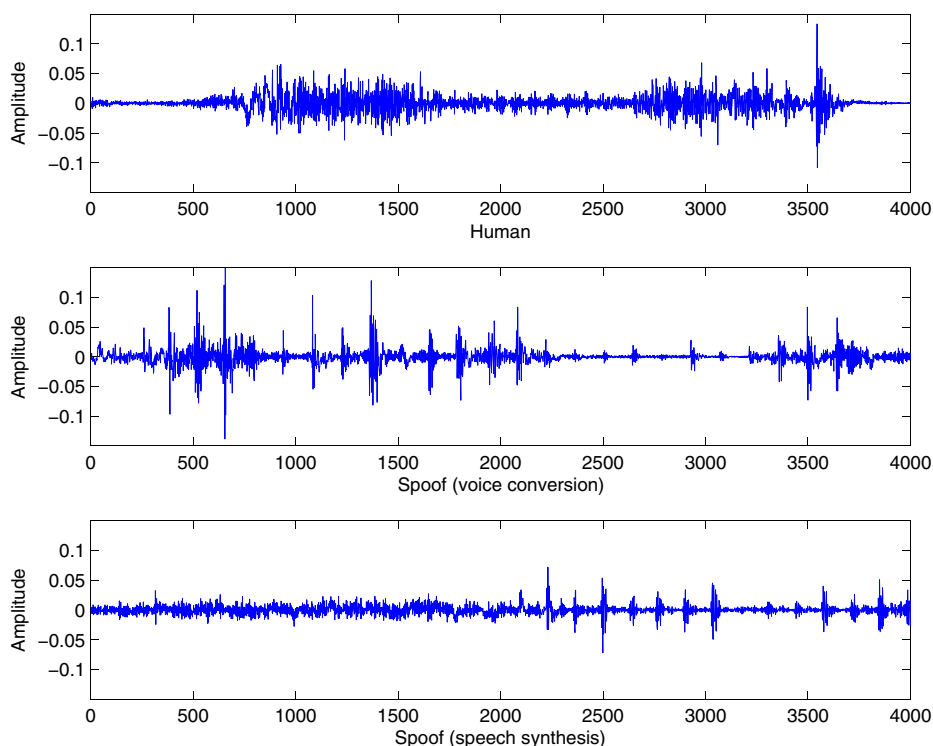
The LPC technique is widely used in speech coding, e.g., in GSM 06.10 [11] or in narrow-band and wide-band adaptive multi-rate coders (AMR) [6]. It can also be used to parametrize signals in speech or speaker recognition. Linear prediction can also be applied to vectors, and in such a case, a vector of samples is predicted using another vector of samples from the signal's history. This method is called long-term prediction (LTP) in contrast to LPC, sometimes referred to as short-term prediction. LTP is often used on top of LPC, i.e., the LPC error is further processed by LTP. This approach has been encountered in [11], for example. LTP works especially well for voiced speech, where the signal is quasi-periodic, so this makes it easier to match a similar vector in the past. Prediction error and prediction gain for LTP are defined analogously as for LPC.

### 3 Proposed spoofing countermeasure

The proposed ASV spoofing countermeasure is based on analysis of the prediction error in the speech signal at the ASV input. One may expect that if a non-natural speech signal undergoes the prediction process, it may be either “too well” predicted (i.e., with a high prediction gain) or ineffectively predicted (i.e., with a prediction gain lower than usual).

Figure 1 shows three residual (error) signals left from prediction processing of voiced speech. It can be observed that the prediction error of a spoof signal generated with voice conversion (middle) exhibits sudden bursts of errors, probably due to the non-smooth frame concatenation used (in this case the voice conversion algorithm was based on frame selection). These bursts are separated by a low energy noise-like prediction error signal, which may imply a much more efficient prediction than for natural human speech (top figure). The prediction error of the spoof signal generated with speech synthesis (bottom figure) is also much weaker and less dynamic than for natural speech. Therefore, the proposed spoofing countermeasure involves the extraction of various parameters of prediction error, hoping to capture the features that will help to differentiate genuine from spoof speech signals.

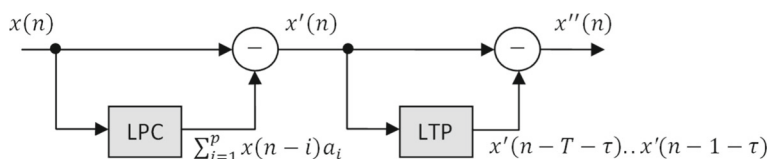
The proposed speech processing process is shown in Fig. 2. It is similar to the speech coding process used, e.g., in GSM 06.10 coding [11]. The input signal  $x(n)$  is first analyzed using the LPC technique, where the  $p$  prediction coefficients  $a_i$  are estimated. The predicted



**Fig. 1** Residual signal ( $x''(n)$ ) at the output of the LTP block for human speech (*top*), spoof signal S1 generated with voice conversion (*middle*) and spoof signal S3 generated with speech synthesis (*bottom*), all for voiced speech. All signals originated from the ASVspoof 2015 corpus [41] and are described in detail in Section 4

values are subtracted from the original samples, and the resulting LPC prediction error signal  $x'(n)$  is processed further. The next block, LTP, operates on vectors of samples rather than on individual samples. When the best matching vector is found, it is subtracted from signal  $x'(n)$ , which results in the LTP prediction error signal  $x''(n)$ .

It must be stressed that even though the linear prediction technique described above is well-known, the way it is used in the proposed method is significantly novel. In the classical application of linear prediction (e.g., in the GSM 06.10 speech coding), the main effort is invested in minimizing the prediction error, so that the majority (in this case: 2/3) of the residual (error) samples can be zeroed. In contrast, the prediction coefficients  $a_i$  are



**Fig. 2** Schematic picture of speech processing in the proposed countermeasure

transmitted further and well protected against transmission errors, because they convey the majority of the acoustic information. However, in the proposed countermeasure the prediction coefficients are completely disregarded. In contrast, the utmost attention is paid to the prediction error, as it is claimed that these data carry information about the nature of the speech signal, i.e., whether it is genuine or artificial.

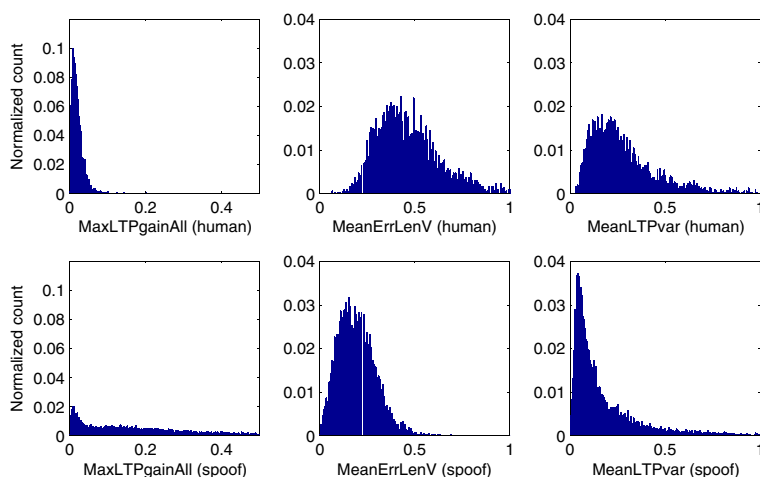
In order to extract this information, in the proposed approach, the energy of prediction errors resulting from both blocks and the ratios between them (i.e., LTP prediction gain) are analyzed. In addition, the temporal distribution of LTP errors is analyzed by measuring the length of segments with a prediction error above a certain level. In total, 23 parameters are extracted, which can be divided into three groups:

- Eight parameters related to the LPC error:
  - *MeanLPCerrAll* – mean energy of the LPC error for the whole signal, i.e., mean energy of  $x'(n)$ ;
  - *MaxLPCerrAll* – maximum energy of the LPC error for the whole signal;
  - *MeanLPCerrV* – mean energy of the LPC error narrowed down to the voiced regions;
  - *MaxLPCerrV* – maximum energy of the LPC error for the voiced speech;
  - *MeanLPCgainAll* – mean LPC gain for the whole signal (i.e., mean ratio between the energies of the input signal and the LPC error, mean  $G_p$  as defined in (2));
  - *MaxLPCgainAll* – maximum LPC gain for the whole signal;
  - *MeanLPCgainV* – mean LPC gain narrowed down to the voiced regions;
  - *MaxLPCgainV* – maximum LPC gain narrowed down to the voiced regions.
- Ten energy-related parameters of the LTP error:
  - *MeanLTPerrAll* – mean energy of the LTP error for the whole signal, i.e., mean energy of  $x''(n)$ ;
  - *MaxLTPerrAll* – maximum energy of the LTP error for the whole signal;
  - *MeanLTPerrV* – mean energy of the LTP error narrowed to voiced regions;
  - *MaxLTPerrV* – maximum energy of the LTP error for voiced regions;
  - *MeanLTPgainAll* – mean LTP gain (i.e., mean ratio between energies of the LPC and LTP errors);
  - *MaxLTPgainAll* – maximum of the LTP gain;
  - *MeanLTPgainV* – mean LTP gain for voiced speech;
  - *MaxLTPgainV* – maximum LTP gain for voiced speech;
  - *MeanLTPvar* – mean variance of the LTP gain error for the whole signal;
  - *MaxLTPvar* – maximum variance of the LTP gain error for the whole signal.
- Five time-related parameters of the LTP error:
  - *MeanErrLen* – mean length of segments with LTP error above threshold  $\theta$ ;
  - *MaxErrLen* – maximum length of segments with LTP error above threshold  $\theta$ ;
  - *MeanNoErrLen* – mean length of segments with LTP error equal to or below threshold  $\theta$ ;
  - *MaxNoErrLen* – maximum length of segments with LTP error equal to or below threshold  $\theta$ ;
  - *ErrChangeRate* – LTP threshold crossing rate (counted per 20ms frame).

Time-related parameters of the LTP error were calculated for the whole signal (including the regions with no speech activity), as it was suspected that various speech synthesis artifacts can be visible throughout the whole signal. The initial version of this algorithm, proposed by [19], used only a 10-element subset of the parameters presented above, mainly the ones calculated for voiced speech. This version of the anti-spoofing algorithm will hereinafter be referred to as LPAv1 (linear prediction analysis version 1).

Comparing with LPAv1, in its new version, the parameters calculated for the whole signal were added, as well as the parameters related to LPC gain and the parameters calculated based on the variance of the LTP error. The spoofing countermeasure using the full set of 23 parameters proposed in this study will hereinafter be referred to as LPAv2 (linear prediction analysis version 2).

Figure 3 shows the distributions of the three selected normalized parameters: MaxLTPgainAll, MaxErrLenV and MeanLTPvar, for both human and spoof trials. It is noteworthy that MaxLTPgainAll and MeanLTPvar are the parameters newly added to the improved LPAv2 version of the algorithm. The distributions show that for spoof trials MaxLTPgainAll reaches much higher values than for natural speech; the probability distribution for the spoof trials is close to uniform, while for natural speech the lowest values are by far more probable than the higher ones. It is believed that such distributions are caused by the higher determinism of synthetic and converted voices. For the same reason, MaxErrLenV values for human trials are higher – the prediction errors for human speech not only have higher values, but these high error values also last longer. In contrast, for spoof trials they often take the form of short-term bursts caused by synthesis artifacts, and are separated by low energy errors. The third parameter shown in Fig. 3 – MeanLTPvar – is also usually higher for human speech, because the synthetic speech generation (typical for speech synthesis or voice conversion) usually results in more steady values for the long-term prediction error, and thus it returns a lower variance. These clearly visible differences in distributions of the parameters, extracted based on linear prediction, suggest that spoofing detection using these parameters can be effective. This is going to be verified experimentally in the following sections.



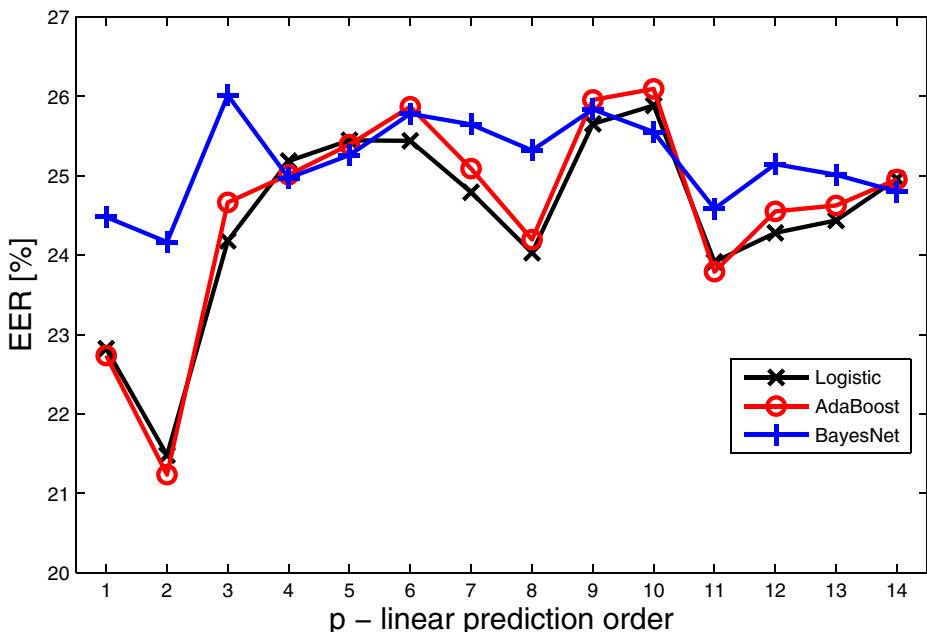
**Fig. 3** Histograms of selected features for human (*top*) and spoof (*bottom*) trials, for all speech samples in the Training dataset (S2 excluded). Scale is preserved between the upper and lower row.

## 4 Experimental set-up

The experiments were conducted on the corpus provided by the ASVspoof 2015 Challenge organizers [41]. The corpus, originating from the English Multi-speaker Corpus for CSTR Voice Cloning Toolkit, contained recordings of various access trials, annotated either as human voice or as a spoof trial. The recordings were divided into three parts: Training, Development and Evaluation, and consisted of 16,375, 53,372 and 193,404 recordings, respectively. The spoof trials were generated using 10 different spoofing algorithms (S1..S10), based either on speech synthesis (S3, S4, and S10) or on voice conversion (the remaining ones). Their spoofing efficiency ranged from 25.42 % to 45.79 % EER, with the exception of S2, which yielded a very low spoofing efficiency equal to 0.87 % EER. The baseline EER value achieved with the iVectors-PLDA system equaled 0.42 % [41].

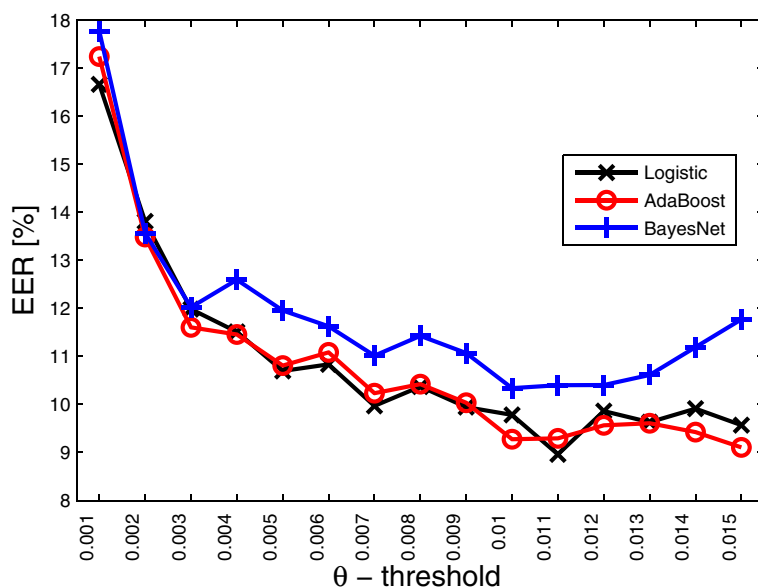
The proposed spoof detection system and baseline LBP-based detector were trained using the Training database. Experiments with spoofing detection, including parameter tuning, were run using the Development corpus. Since the spoofing efficiency caused by method S2 was very low, some of the measurements were done on a subset of recordings without S2 trials. These recordings were also excluded from the training set because preliminary experiments showed that they had a negative impact on the spoofing detection. The Evaluation corpus was tested to show the performance both for already known and for previously unseen spoofing algorithms.

To increase granularity, both errors were calculated sample-by-sample and not frame-wise, which is normally the case in speech coding. Prediction order  $p$  and threshold value  $\theta$  were set heuristically by swapping the values within a certain range, as shown in Figs. 4 and 5, respectively. As a result,  $p$  was set to two and  $\theta$  was set to 0.011, as all the classifiers returned the lowest EER results for these values. The prediction order of two



**Fig. 4** Spoofing detection performance (EER in percentages) against the LPC order, for various classifiers





**Fig. 5** Spoofing detection performance (EER in percentages) against the LTP error threshold value, for various classifiers

is a low value and is unlikely to happen in speech coding. We suspect, however, that such a low order retains the LPC prediction error at a quite high level, and this apparently turns out to be beneficial in further error processing. This case confirms that the proposed way of using linear prediction is significantly different from the classical way that linear prediction is applied, in which the main focus is on minimizing the prediction error.

Contrary to previous research [19], the analysis of the prediction error was run not only for voiced regions, but for the whole signal. Wherever the analysis was narrowed to voiced regions, voicing detection was realized using the SWIPE pitch detector [7].

Feature extraction of the baseline LBP-based countermeasure was set up according to the description in [2]. Each signal was analyzed forming a feature matrix consisting of 16 cepstral coefficients plus energy, their deltas and delta-delta coefficients, which was further analyzed using 58 possible uniform LBP patterns. As a result, 2842 features were generated for every recording.

Data analysis showed that the training datasets, obtained using either the LBP or LPA-based classifiers, contained a substantial number (ca. 10 %) of outliers. Initial experiments revealed that their presence had a negative impact on the spoofing detection, therefore it was decided to remove them. A filter based on the interquartile range with outlier factor equal to 3.0 was applied to detect the outliers. All the parameters, before feeding classifiers, were normalized.

A range of binary classifiers was used to try to achieve the largest area under the receiver operating characteristic (ROC) curve. Based on these results, initially three classifiers were selected: Logistic, Bayesian Networks and AdaBoosts. The Logistic classifier used logistic regression with the ridge penalty to classify data [15]. The Bayesian Networks method used a network of probabilistic dependencies between random variables [29]. The AdaBoost algorithm is a meta-classifier that iteratively used other “weak” classifiers, such

**Table 1** EER results (in percentages) of the spoof detection for various parameter groups and various classifiers, for the Development set without S2 algorithm

Parameters	Classifiers		
	Logistic	AdaBoost	SVM-RBF
All (LPAv2)	2.697	3.023	1.262
10 prms. (LPAv1)	2.986	3.268	2.714
LPC-based	21.524	21.890	10.321
LTP-energy	16.808	16.834	8.185
LTP-time	8.812	9.028	8.842

as decision trees, and iteratively boosted their performance by exposing them to previously misclassified items [14].

Since Bayesian Networks yielded the worst results in the previous study [19], they were subsequently replaced by support vector machines (SVM) with a radial basis function (RBF) kernel. SVMs, originally proposed in [34], have been used with various kernels (e.g., linear, polynomial and radial) for data classification in a wide range of studies, e.g., in food categorization [30], speaker recognition [10, 20] or personality traits recognition based on handwriting [16]. The RBF kernel [32] is well-known for its high classification power. Since the SVM-RBF classifier for optimal performance requires a careful adjustment of  $C$  and  $\sigma$  parameters, they were set experimentally using a grid search, similar to that in [31], separately for each tested configuration of the spoof detector.

Feature extraction was carried out in Matlab, using Voicebox<sup>1</sup> as a speech processing library. Experiments with classification were run using the WEKA toolkit [17] and the LibSVM toolkit [8].

## 5 Results

The assessment of anti-spoofing protection was realized independently from any ASV system, i.e., each recording underwent a binary decision process to decide, whether a recording was real human speech or a spoof trial. The actual speaker verification process can take place either before or after the spoofing detection process and is beyond the scope of this research.

Similar to many other studies [12, 22], spoofing detection was evaluated by measuring the EER values. In addition, detection error trade-off (DET) curves were plotted to show the relation between false alarm and miss probabilities. The EER values and DET plots were obtained using the Bosaris toolkit.<sup>2</sup>

Table 1 shows the EER results for the Development set (without S2 algorithm) obtained using various classifiers and different subsets of parameters. It turned out that the best EER result achieved in this study was less than 1.3 %, which is much less than the EER of 3 % obtained using the previous approach [19]. The best result was returned for the SVM-RBF classifier and the complete set of parameters. Any trials with reducing the number of parameters, e.g., using parameter selection, turned out to worsen the results. Table 1 shows that using the SVM-RBF classifier with the set of parameters used in the initial version of

<sup>1</sup><http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html>

<sup>2</sup><https://sites.google.com/site/bosaristoolkit/>

**Table 2** EER results (in percentages) of the spoof detection, for Development and Evaluation datasets, for various classifiers and the tested countermeasures

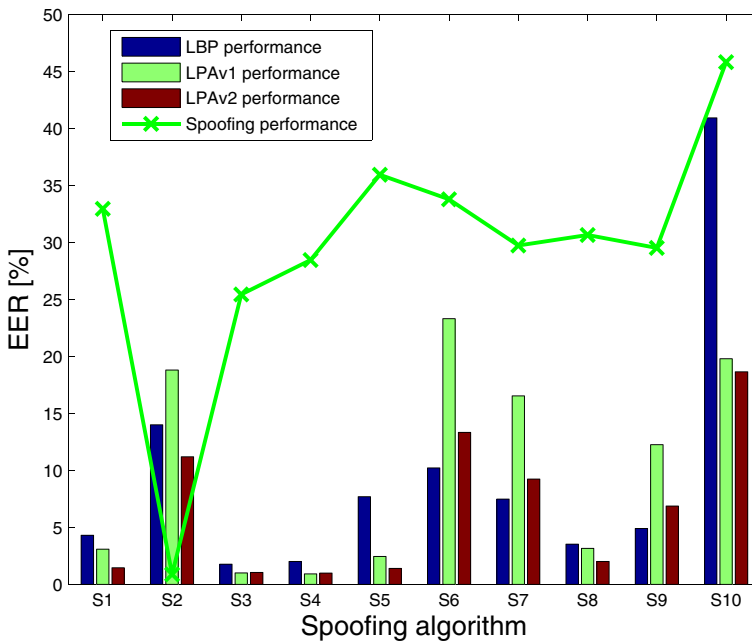
Counter-	Classifiers		
measures	Logistic	AdaBoost	SVM-RBF
	Development (S2 excluded)		
LBP	11.529	8.065	4.336
LPAv1	2.986	3.268	2.714
LPAv2	2.697	3.023	1.262
	Evaluation		
LBP	12.389	13.429	11.480
LPAv1	11.616	14.334	12.647
LPAv2	13.623	14.567	7.950

the anti-spoofing algorithm (LPAv1) only slightly improved the results (3.0 % vs. 2.7 %). When observing the EER results for other groups of parameters one can notice that the LTP-based parameters seem to be of greater importance than the LPC-based ones.

Table 2 compares the results between the two versions of the LPA algorithm and the results using the LBP features. The comparison comprised of three classifiers and two datasets (Development without S2 and the Evaluation sets). For both corpora, the new version of the anti-spoofing algorithm (LPAv2) outperformed both the previous version (LPAv1) and the baseline LBP algorithm. The spoofing detection EER for the LBP-based anti-spoofing for the Development set without S2 reached 4.3 %, while for the proposed algorithm it was as low as 1.3 %. Again, the best results were achieved for the SVM-RBF classifier, with the exception of LPAv1 for the Evaluation set, where the Logistic classifier turned out to be slightly better (11.6 % vs. 12.6 %). It is suspected that the Logistic classifier in the case of LPAv1 and its 10-dimensional space was more resistant to the unseen spoofing algorithms than the SVM-RBF classifier. However, for LPAv2 the SVM-RBF classifier performed best both for the known spoofing algorithms (Development) and the mixed known and new ones (Evaluation).

Figure 6 displays the countermeasure performance for the Evaluation dataset for various spoofing algorithms, as well as the line showing how harmful each of these algorithms was. Unsurprisingly, the algorithms that were present in the Training corpus (S1, S3, S4 and S5) in most of the cases were also detected best by the tested algorithms. Algorithm S4, using speech synthesis with STRAIGHT vocoder [21], was the easiest to detect – the LPAv2 algorithm yielded here less than 1 %. In addition, spoofing algorithm S8, using voice conversion, was detected with high accuracy (EER below 2 %), even though it was not present in the Training database. The remaining spoofing algorithms, not present in the Training dataset, were detected with higher EERs, which is confirmed in Table 3. For example, the S2 algorithm was detected by the proposed method with an EER equal to 11.2 %, but as a matter of fact this number is of low importance, since the spoofing efficiency of this algorithm was as low as 0.9 %. In contrast, a very efficient spoofing algorithm S10, based on the MARY Text-To-Speech system,<sup>3</sup> turned out to be the most difficult to detect, which was

<sup>3</sup><http://mary.dfki.de/>



**Fig. 6** Spoofing performance (solid line) along with LPAv1 and LPAv2 countermeasure performance (left and right bars, respectively) for various spoofing algorithms from the Evaluation dataset

most likely due to a lack of vocoder. However even here, the EER returned by the proposed system equaled 18.6 %, which is much lower than the spoofing efficiency of S10: 45.8 %.

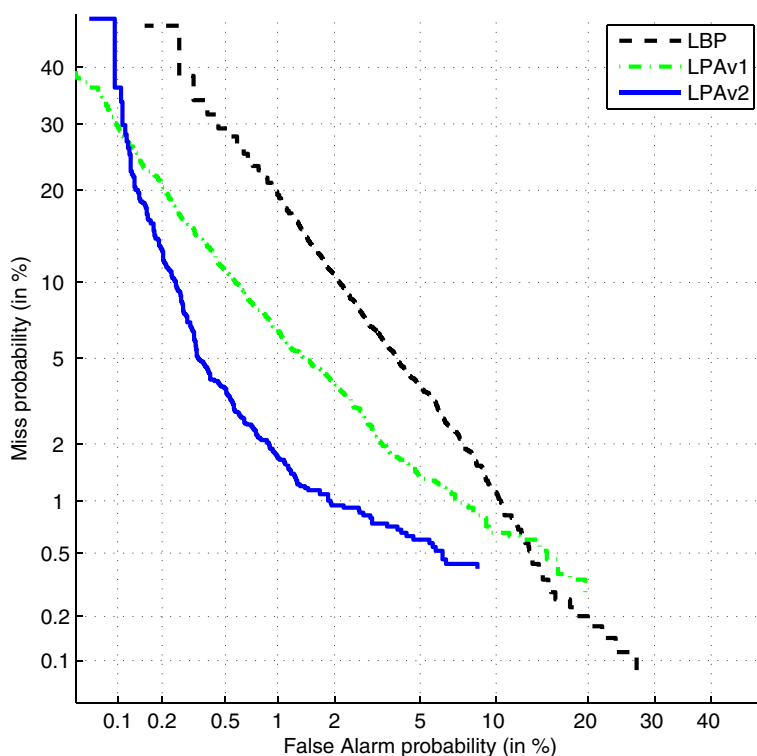
We observed no correlation between the detection results for the spoofing signals generated using different vocoders, i.e., for S5 (using the MLSA vocoder) and, e.g., S4 and S8 (using the STRAIGHT vocoder), which returned EER values of 1.4 %, 1.0 % and 2.0 %, respectively. This suggests that the proposed solution is vocoder-independent.

The anti-spoofing protection proposed in this article outperformed the previous version [19] for most of the spoofing algorithms, apart from S3 and S4, where the results were in fact the same. The difference in performance is especially visible as for the voice conversion-based spoofing methods (see Table 3). The average spoofing detection efficacy for speech synthesis attacks (algorithms: S3, S4 and S10) was almost the same for LPAv1 and LPAv2.

When comparing the LPAv2 with the LBP-based spoofing countermeasure, the LPAv2 results were better both for known and previously unseen spoofing algorithms (EER equaled

**Table 3** Spoofing and countermeasure performance (EER in percentages) for Evaluation dataset, for the LBP, LPAv1 and LPAv2 countermeasures

Spoof.alg.	Spoof.perf.	LBP	LPAv1	LPAv2
Known	30.675	3.941	1.859	1.217
New	28.377	13.496	15.631	10.206
Conversion	27.616	7.438	11.362	6.581
Synthesis	33.217	14.891	7.229	7.240
Average	29.296	9.674	10.122	6.611



**Fig. 7** DET plots for the three analyzed countermeasures, tested on the Development set without S2

1.2 % and 10.2 % vs. 1.9 % and 15.6 %, for the LPAv2 and LBP algorithms, respectively). However, when it came to comparing individual spoofing algorithms, it turned out that for three spoofing algorithms (S6, S7 and S9) the LBP-based countermeasure returned better results. For example, for algorithm S6, the returned EER results were 10.2 % and 13.3 % for LBP and LPAv2, respectively. What is worth noting is that all these algorithms were not present in the training set. In contrast, the LPAv2 anti-spoofing method returned much better results for other unseen algorithms (S2, S8, S10), so at this stage it would be difficult to definitively judge which of these two algorithms has better generalization capabilities.

The DET plots, shown in Fig. 7 for the Development corpus without S2, confirmed the good spoofing detection capabilities of the proposed LPAv2 algorithm. The DET curve of LPAv2 is convex and closest to the coordinate origin, while the DET curves for the LBP and LPAv1 countermeasures are close to straight, and placed at a larger distance from the (0,0) point. The DET plot for LPAv2 indicates, however, that the proposed countermeasure was not able to decrease the miss probability below ca. 0.5 %.

## 6 Conclusions and future work

This article presented a new version of the algorithm that aimed to increase the anti-spoofing protection of speaker verification systems against unauthorized access using speech synthesis, voice conversion and, potentially, other attacks. The results shown in this paper were

compared with the previous version of this algorithm (LPAv1) as well as with the LBP-based anti-spoofing method.

The proposed spoofing countermeasure is based on analysis of prediction error, which results from cascaded LPC and LTP blocks. As the LP-based vocoder is often a part of speech synthesis or voice conversion systems, by using linear prediction analysis in some sense a reverse operation is performed, to verify if vocoding really took place.

The described new version of the countermeasure, called LPAv2, uses an extended set of parameters – 23 instead of 10, including LPC gain and variance of LTP error, as well as the parameters extracted from the whole speech signal (and not only from the voiced speech as in LPAv1). The new version employs a more powerful classifier (SVM-RBF), in which the parameters were tuned to maximize the detection efficiency.

It turned out that the proposed version, with an enlarged number of parameters and the SVM-RBF classifier, was able to improve the EER results by lowering them from 3 % for LPAv1 down to 1.3 %, when using the Development corpus without S2. The analyses shown in this study suggest that the LTP-based parameters contributed most to successful spoofing detection.

In addition, the results for the Evaluation dataset were better: 8 % for LPAv2 vs. 11.6 % for LPAv1. Some scientists have reported better results here, e.g., [18] used a classical GMM-based system that returned an EER of 3 %. However, a huge difference between the performance for the known attacks (1.2 % EER) and the unknown ones (10.2 % EER) may imply that the proposed algorithm requires further parameter tuning to be able to better detect previously unseen spoofing algorithms.

In most of the cases, the proposed method performed better than a baseline LBP-based detector. It is likely that the LBP detector required longer speech data, as in [2] it was tested on 5 min. recordings, while the recordings tested in the current study were no longer than several seconds. For three out of 10 spoofing algorithms present in the Evaluation dataset, the results using the LBP-based features and the SVM-RBF classifier outperformed the proposed method. It would be interesting to know if the fusion of scores returned by the LBP and LPAv2 countermeasures would lead to an increase in the spoofing detection – this may be a subject for future work.

It is hoped that the improved version of the spoofing countermeasure, based on an analysis of prediction error, as well as the analyses presented in this article, will help to increase the security of speaker verification systems. Future work can focus on elaborating a generalized countermeasure able to precisely detect a wide range of spoofing attacks against ASV systems, in which the proposed algorithm can be one of the contributing elements.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

1. Alegre F., Amehraye A., Evans N. (2013) Spoofing countermeasures to protect automatic speaker verification from voice conversion. In: Proc. IEEE int. Conf. Acoust., speech and signal process. (ICASSP)
2. Alegre F., Vipperla R., Amehraye A., Evans N. (2013) A new speaker verification spoofing countermeasure based on local binary patterns. In: Proc. Interspeech. Lyon, France, p 2013

3. Alegre F., Janicki A., Evans N. (2014) Re-assessing the threat of replay spoofing attacks against automatic speaker verification. In: Proc. 13th international conference of the biometrics special interest group (BIOSIG). Darmstadt, Germany, pp 157–168
4. Australian Government (2014) ATO launches voice authentication. <https://www.ato.gov.au/media-centre/media-releases/ato-launches-voice-authentication/>
5. Bank Smart (2015) Bank SMART - biometria. <http://www.banksmart.pl/aplikacja/biometria/>
6. Bessette B., Salami R., Lefebvre R., Jelinek M., Rotola-Pukkila J., Vainio J., Mikkola H., Järvinen K. (2002) The adaptive multirate wideband speech codec (AMR-WB). *IEEE Trans Speech and Audio Process* 10(8):620–636
7. Camacho A. (2007) SWIPE: A Sawtooth waveform inspired pitch estimator for speech and music. Ph.D. thesis, Gainesville, FL USA
8. Chang C. C., Lin C. J. (2011) LIBSVM: A library for support vector machines. *ACM Trans Intell Syst Technol* 2:27:1–27:27
9. De Leon P. L., Hernaez I., Saratxaga I., Pucher M., Yamagishi J. (2011) Detection of synthetic speech for the problem of imposture. In: Proc. IEEE int. Conf. Acoust., speech and signal process. (ICASSP), pp 4844–4847
10. Ding I. J., Yen C. T. (2015) Enhancing GMM speaker identification by incorporating SVM speaker verification for intelligent web-based speech applications. *Multimedia Tools and Appl* 74(14):5131–5140. doi:10.1007/s11042-013-1587-5
11. ETSI (1999) Digital cellular telecommunications system (Phase 2+) (GSM); Full rate speech; Transcoding GSM 06.10
12. Evans N., Kinnunen T., Yamagishi J. (2013) Spoofing and countermeasures for automatic speaker verification. In: Proc. Interspeech. Lyon, France, p 2013
13. Farrús M., Wagner M., Anguita J., Hernando J. (2008) How vulnerable are prosodic features to professional imitators? In: Proc. Odyssey. ISCA, Stellenbosch, South Africa
14. Freund Y., Schapire R. E. (1999) A short introduction to boosting. In: Proceedings of the sixteenth international joint conference on artificial intelligence, pp. 1401–1406. Morgan kaufmann
15. Friedman J. H., Hastie T., Tibshirani R. (2010) Regularization paths for generalized linear models via coordinate descent. *J Stat Softw* 33(1):1–22
16. Górska Z., Janicki A. (2012) Recognition of extraversion level based on handwriting and support vector machines. *Perceptual & Motor Skills* 114(3):857–869
17. Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., Witten I. H. (2009) The WEKA data mining software: An update. *SIGKDD Explor Newsl* 11(1):10–18
18. Haniłçi C., Kinnunen T., Sahidullah M., Sizov A. (2015) Classifiers for synthetic speech detection: A comparison. In: Proc. Interspeech 2015. Dresden, Germany, pp 2057–2061
19. Janicki A. (2015) Spoofing countermeasure based on analysis of linear prediction error. In: Proc. Interspeech 2015. Dresden, Germany, pp 2077–2081
20. Janicki A., Staroszczyk T. (2011) Speaker recognition from coded speech using support vector machines. In: Proceedings of the 14th international conference on text, speech and dialogue, TSD'11. Springer, Berlin, pp 291–298
21. Kawahara H., Masuda-Katsuse I., de Cheveigné A. (1999) Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds. *Speech Comm* 27(3):187–207
22. Kinnunen T., Wu Z., Lee K. A., Sedlak F., Chng E. S., Li H. (2012) Vulnerability of Speaker Verification Systems Against Voice Conversion Spoofing Attacks: The case of Telephone Speech. In: Proc. IEEE int. Conf. Acoust., speech and signal process. (ICASSP), pp 4401–4404
23. Lau Y. W., Wagner M., Tran D. (2004) Vulnerability of speaker verification to voice mimicking. In: Proc. International symposium on intelligent multimedia, video and speech processing, pp 145–148. IEEE, Hong Kong, China
24. Lindberg J., Blomberg M. (1999) Vulnerability in speaker verification - a study of technical impostor techniques. In: European conference on speech communication and technology, pp 1211–1214
25. Makhoul J. (1975) Linear prediction: A tutorial review. *Proc IEEE* 63(4):561–580
26. Masuko T., Hitotsumatsu T., Tokuda K., Kobayashi T. (1999) On the security of HMM-based speaker verification systems against imposture using synthetic speech. In: Proc. EUROSPEECH
27. Ogihara A., Shiozaki A. (2005) Discrimination method of synthetic speech using pitch frequency against synthetic speech falsification. *IEICE Trans Fundam Electron Commun Comput Sci* 88(1):280–286
28. Ojala T., Pietikainen M., Maenpää T. (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Trans Pattern Anal Mach Intell* 24(7):971–987
29. Pearl J. (1988) Probabilistic reasoning in intelligent systems: Networks of plausible inference. Morgan Kaufmann publishers Inc., San Francisco

30. Pouladzadeh P., Shirmohammadi S., Bakirov A., Bulut A., Yassine A. (2015) Cloud-based SVM for food categorization. *Multimedia Tools Appl* 74(14):5243–5260. doi:[10.1007/s11042-014-2116-x](https://doi.org/10.1007/s11042-014-2116-x)
31. Rybka J., Janicki A. (2013) Comparison of speaker dependent and speaker independent emotion recognition. *Appl Math Comput Sci* 23(4):797–808
32. Schölkopf B., Sung K. K., Burges C. J., Girosi F., Niyogi P., Poggio T., Vapnik V. (1997) Comparing support vector machines with gaussian kernels to radial basis function classifiers. *IEEE Trans Signal Process* 45(11):2758–2765
33. Vaidyanathan P. P. (2007) *The Theory of Linear Prediction*. Synthesis Lectures on Signal Processing Morgan & Claypool Publishers
34. Vapnik J. N. (1995) *The nature of statistical learning theory*. Springer
35. Villalba J., Lleida E. (2010) Speaker verification performance degradation against spoofing and tampering attacks. In: *FALA Workshop*, pp 131–134
36. Villalba J., Lleida E. (2011) Preventing replay attacks on speaker verification systems. In: *Security Technology (ICST)*, 2011 IEEE International Carnahan Conference on, pp 1–8. doi:[10.1109/CCST.2011.6095943](https://doi.org/10.1109/CCST.2011.6095943)
37. Wang Z. F., Wei G., He Q. H. (2011) Channel pattern noise based playback attack detection algorithm for speaker recognition. In: *Machine Learning and Cybernetics (ICMLC)*, 2011 International Conference on, vol 4, pp 1708–1713. doi:[10.1109/ICMLC.2011.6016982](https://doi.org/10.1109/ICMLC.2011.6016982)
38. Wu Z., Chng E., Li H. (2013) Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition. In: *Proc. 13th Interspeech*, Lyon
39. Wu Z., Larcher A., Lee K. A., Chng E., Kinnunen T., Li H. (2013) Vulnerability evaluation of speaker verification under voice conversion spoofing: the effect of text constraints. In: Bimbot F., Cerisara C., Fougerson C., Gravier G., Lamel L., Pellegrino F., Perrier P. (eds) *INTERSPEECH*, pp. 950–954. ISCA
40. Wu Z., Evans N., Kinnunen T., Yamagishi J., Alegre F., Li H. (2014) Spoofing and countermeasures for speaker verification: A survey. *Speech Comm* 66:130–153. doi:[10.1016/j.specom.2014.10.005](https://doi.org/10.1016/j.specom.2014.10.005)
41. Wu Z., Kinnunen T., Evans N., Yamagishi J., Hanile C., Sahidullah M., Sizov A. (2015) *ASVSpooF* 2015: The first automatic speaker verification spoofing and countermeasures challenge. In: *Proc. Interspeech*. Dresden, Germany, p 2015



**Artur Janicki** received MSc and PhD (1997 and 2004, respectively, both with honors) in telecommunications from the Faculty of Electronics and Information Technology, Warsaw University of Technology (WUT). Assistant Professor at the Institute of Telecommunications, WUT. His research and teaching activities focus on speech processing, including speaker recognition, speech coding and synthesis, emotion recognition, with elements of data mining and information theory. Author or co-author of over 50 conference and journal papers, supervisor of over 40 bachelor and master theses. Member of the International Speech Communication Association (ISCA) and the European Association for Signal Processing (EURASIP).